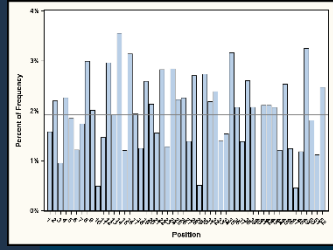
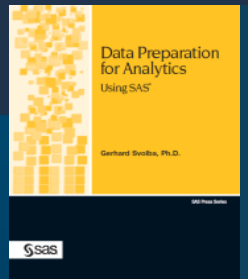


Gewinnen beim Monopoly® Spiel – Alles nur Zufall? Oder gibt es doch ein paar Muster, die man kennen sollte?

21. KSFE, Krefeld, 9.-10. März 2017

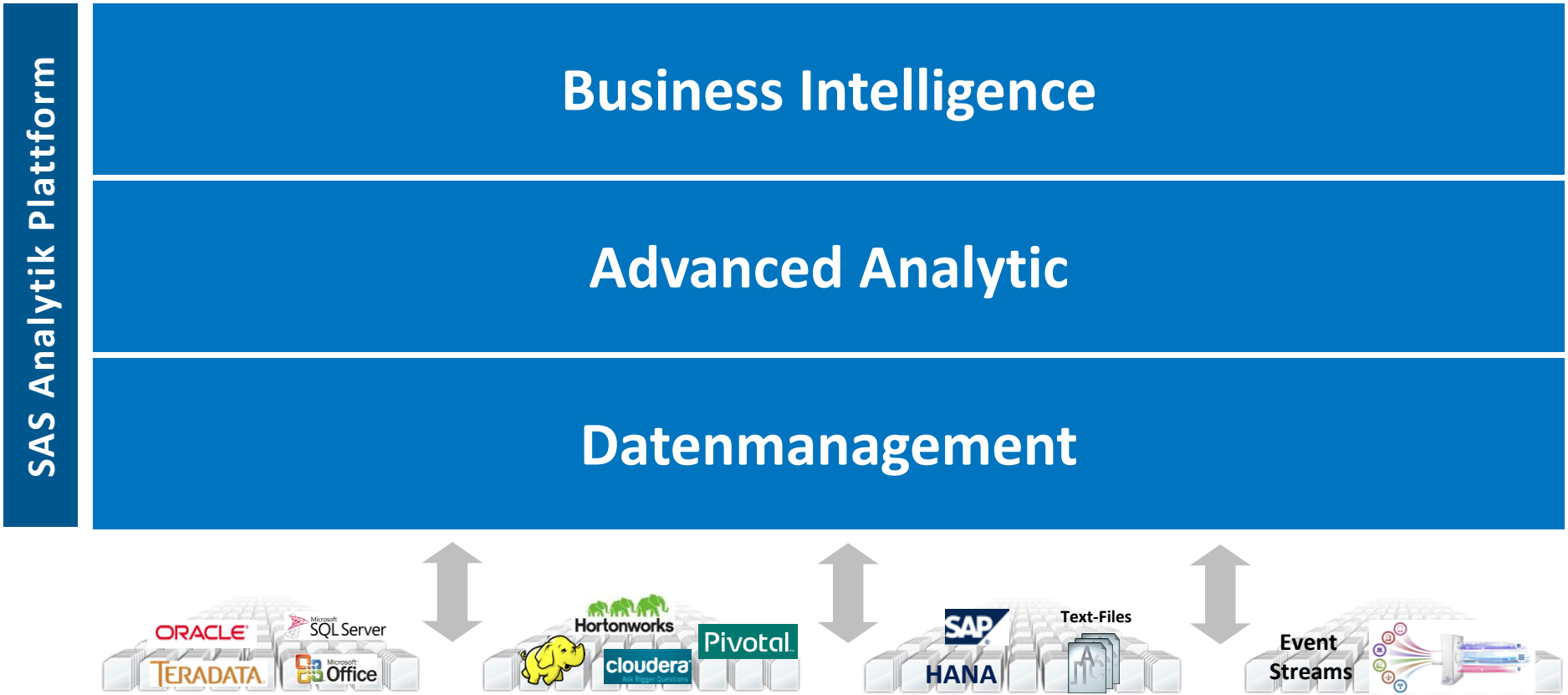
Gerhard Svolba



Die Vortragsfolien sind online
→ Google: Gerhard SAS Samples

SAS Analytik Plattform

Unterschiedliche Layer aus konzeptioneller Sicht



SAS Analytik Plattform

Advanced Analytic Layer

SAS Analytik Plattform

Business Intelligence

Data Mining

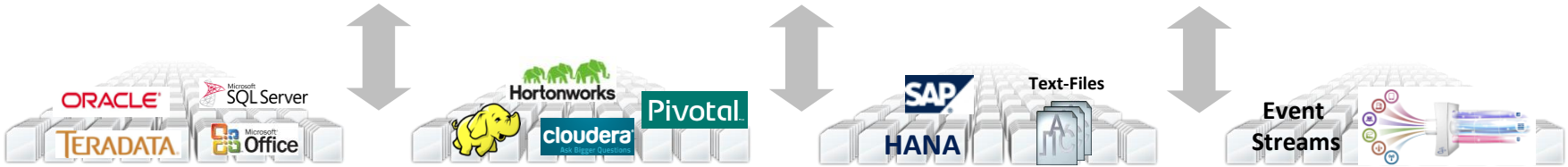
Statistical Analysis

Forecasting

Text Analytics

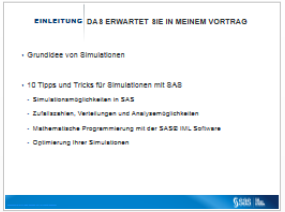
Optimization & Simulation

Datenmanagement

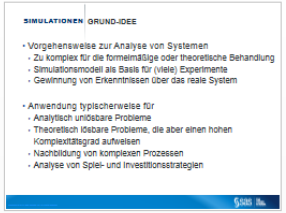


KSFE 2016: **Simulationen und Mathematische Programmierung mit SAS**

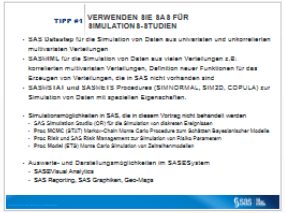
1 ★



2 ★

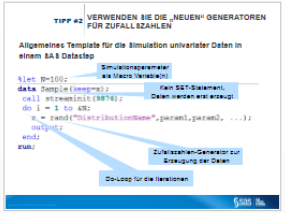


3 ★

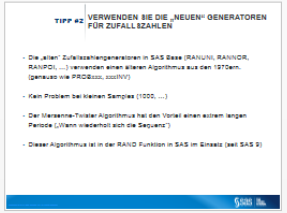


4 ★

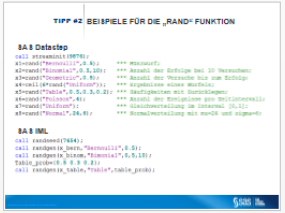
5 ★



6 ★

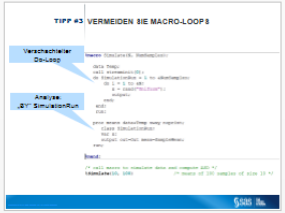


7 ★



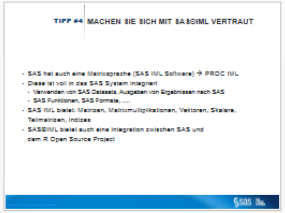
8 ★

9 ★

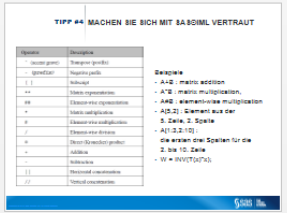


10 ★

11 ★

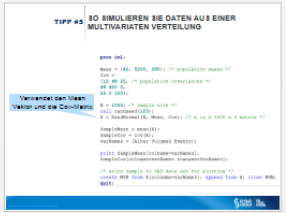


12 ★

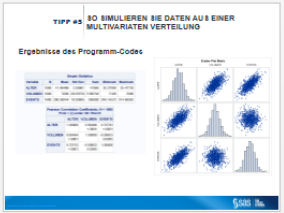


13

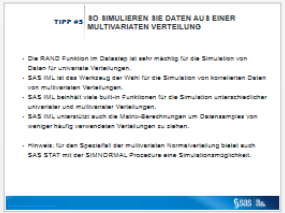
14



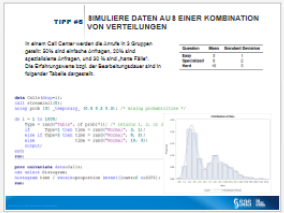
15



16

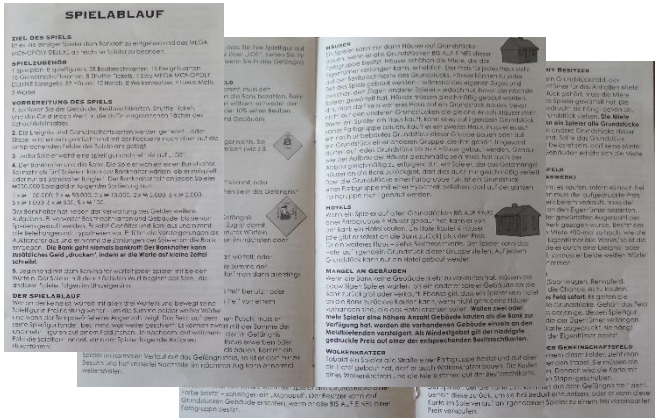


17

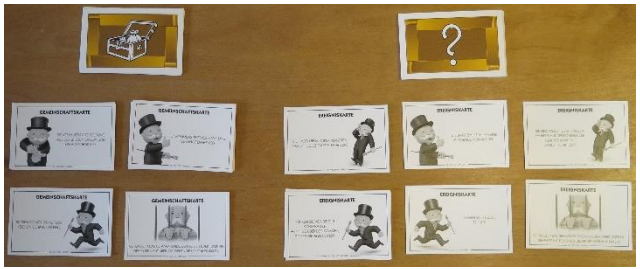


18

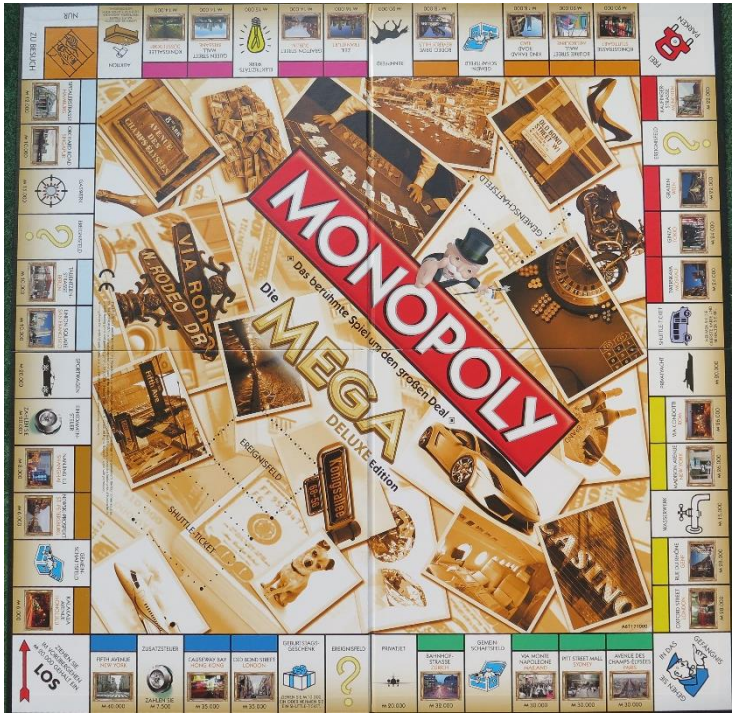
Das Monopoly Spiel ist ein komplexes System



Komplexe Regeln



Zusätzliche Anweisungen



Rahmenwerk von Möglichkeiten und Ereignissen

Monetäre Dimension

Dynamische Komponenten

Zufällige Komponenten

Wichtige Fragestellungen

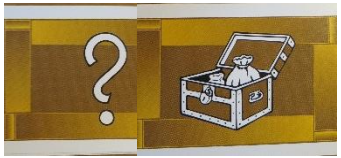
- Wie ist die Verteilung der Besuchshäufigkeiten auf die Felder am Brettspiel?
- Welche Felder sind am profitabelsten?
- Welche Felder haben die höchste Variabilität im Profit?

- **Diese Fragen können auf viele andere Simulationsstudien von komplexen Fragestellen übergeleitet werden.**

Platzierung der Spielfigur - Einflussgrößen

Würfel-
Summe

Gehe ins
Gefängnis!



Ereignis-
Felder

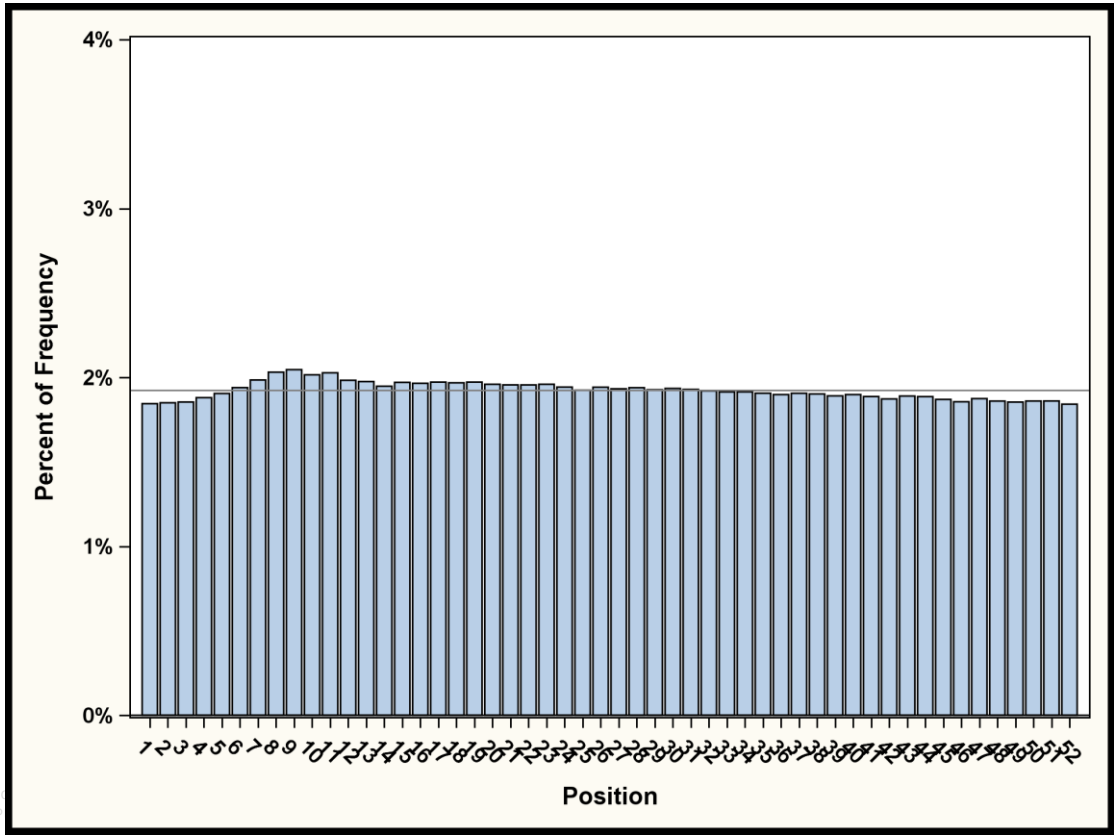
Speed-
Würfel

2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						

Nur Würfeln - Fast eine Gleichverteilung

10000 Simulationen, 70 Runden, 4 Spieler

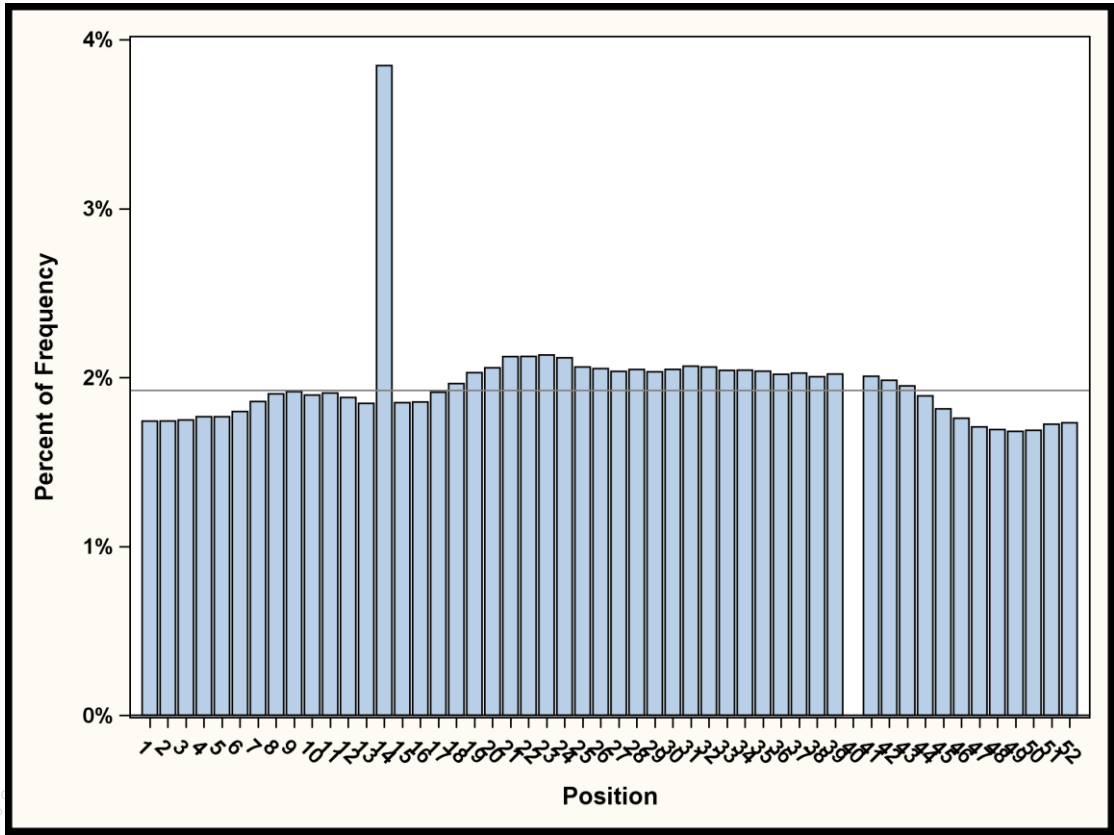
Würfel-Summe



Alle Feld-40 Besuche werden auf Feld 14 umgesetzt

Würfel-Summe

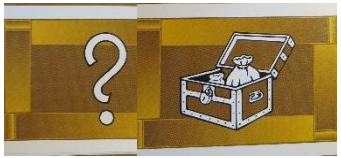
Gehe ins Gefängnis!



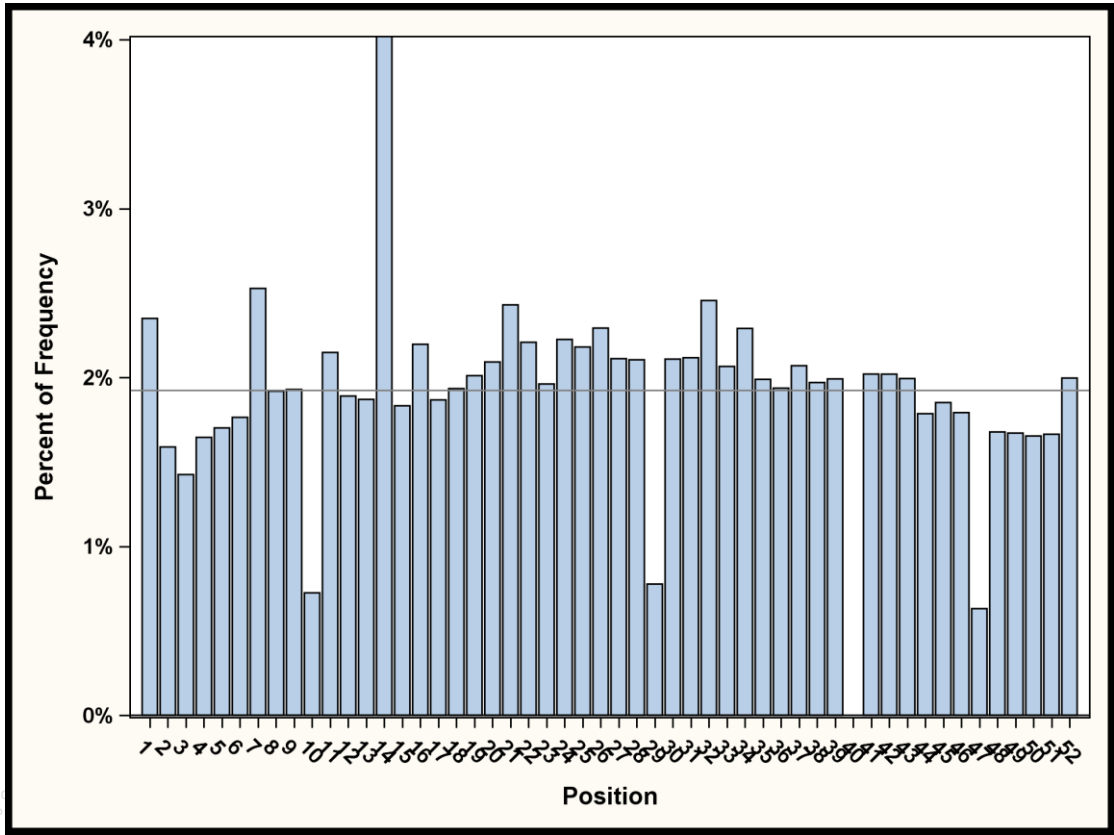
Ereignisfelder positionieren die Spielfigur um

Würfel-
Summe

Gehe ins
Gefängnis!



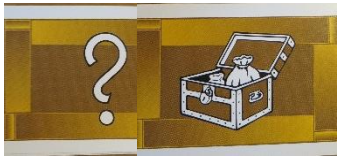
Ereignis-
Felder



Der Speed-Würfel erzeugt zusätzliche Variabilität

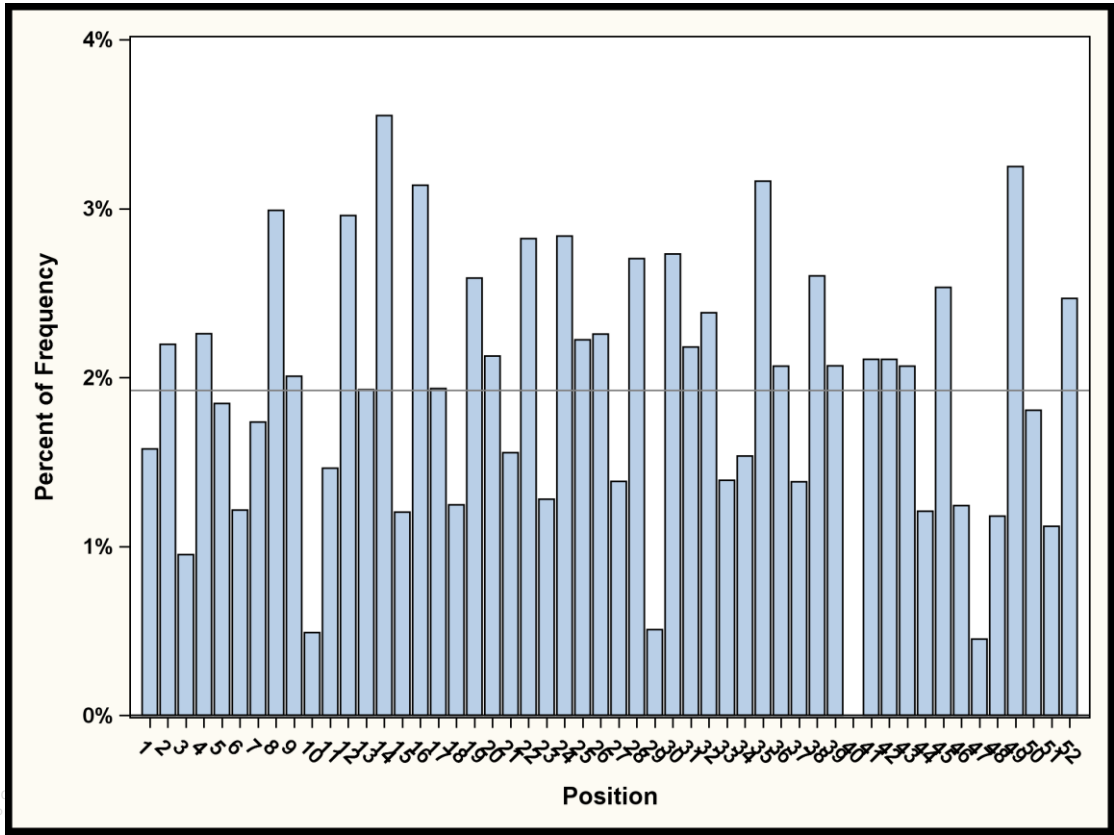
Würfel-Summe

Gehe ins Gefängnis!



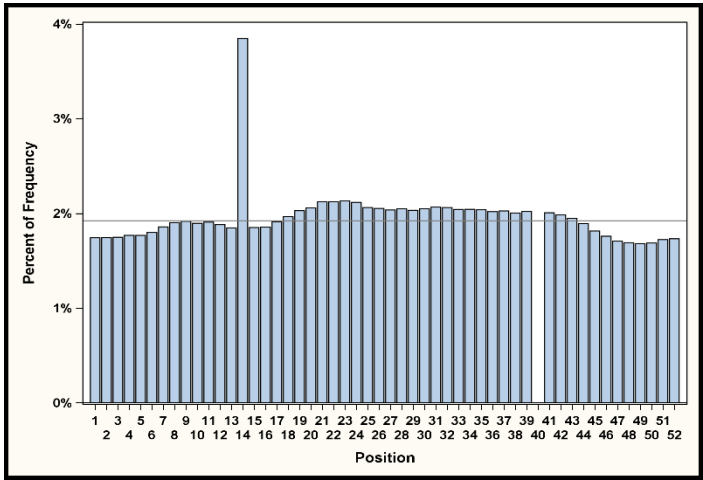
Ereignis-Felder

Accelerator Dice



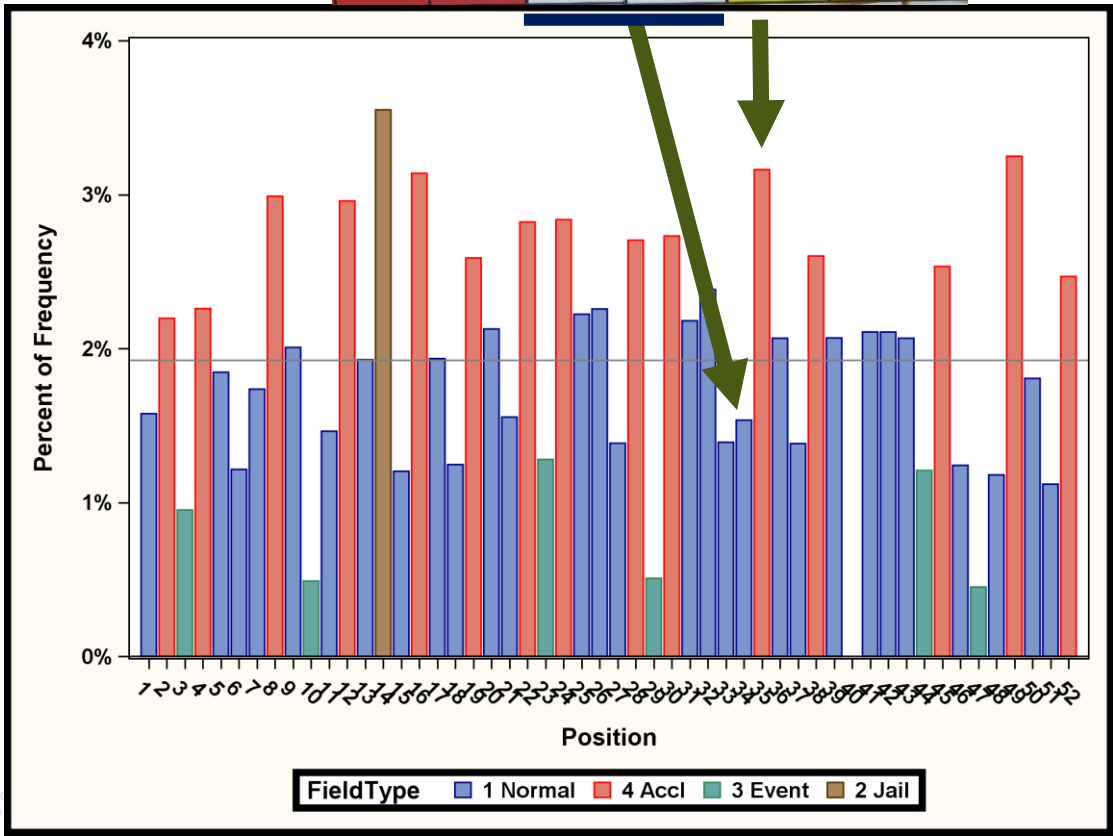
SAS Code für das Histogramm

```
proc sgplot data=PlayerPosition_MNP1;
 vbar Position / stat=percent_MNP1;
 yaxis max=0.04;
 xaxis values=(1 to 52 by 1) fitpolicy=stagger;
 refline 0.01923 /axis=y;
run;
```

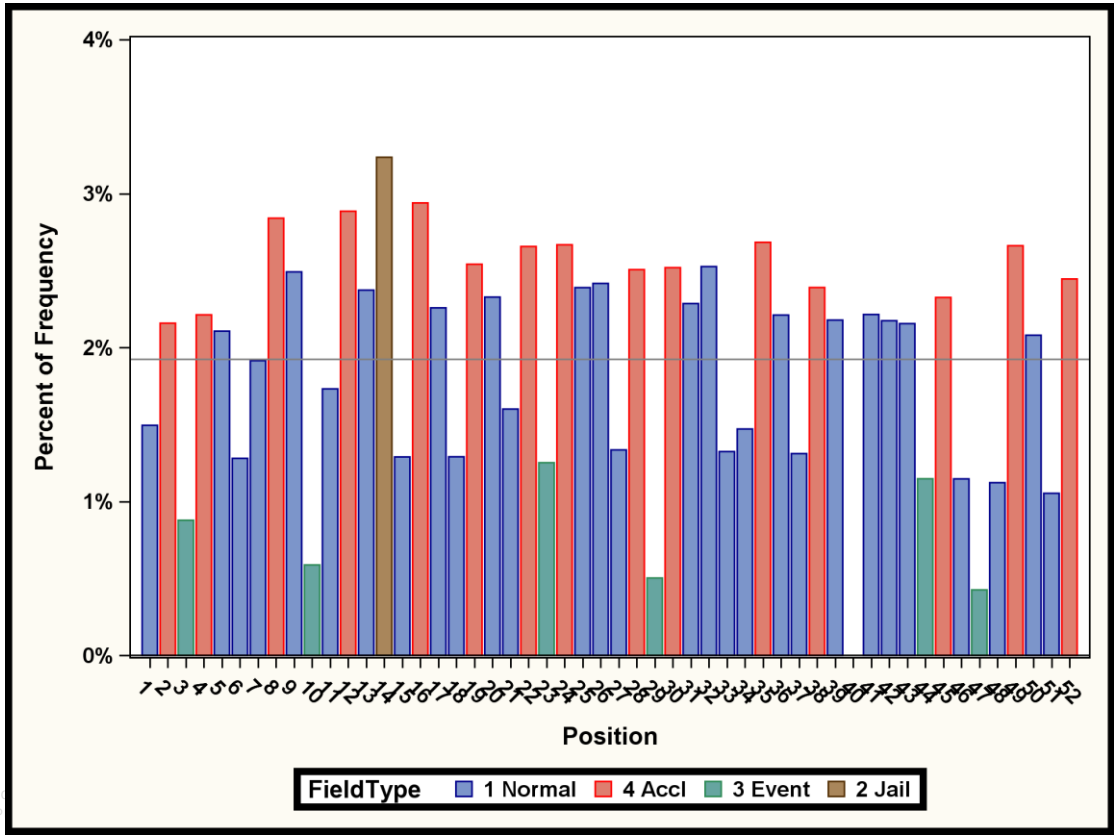


Beispiel für eine Umplatzierung

- Wenn der Speed-Würfel den Monopoly-Mann zeigt:
  - Ziehe weiter bis zum nächsten freien Grundstück
  - Ziehe weiter zum nächsten Grundstück, wenn schon alle verkauft sind.



Effekt des Speedwürfels nach 20 Runden

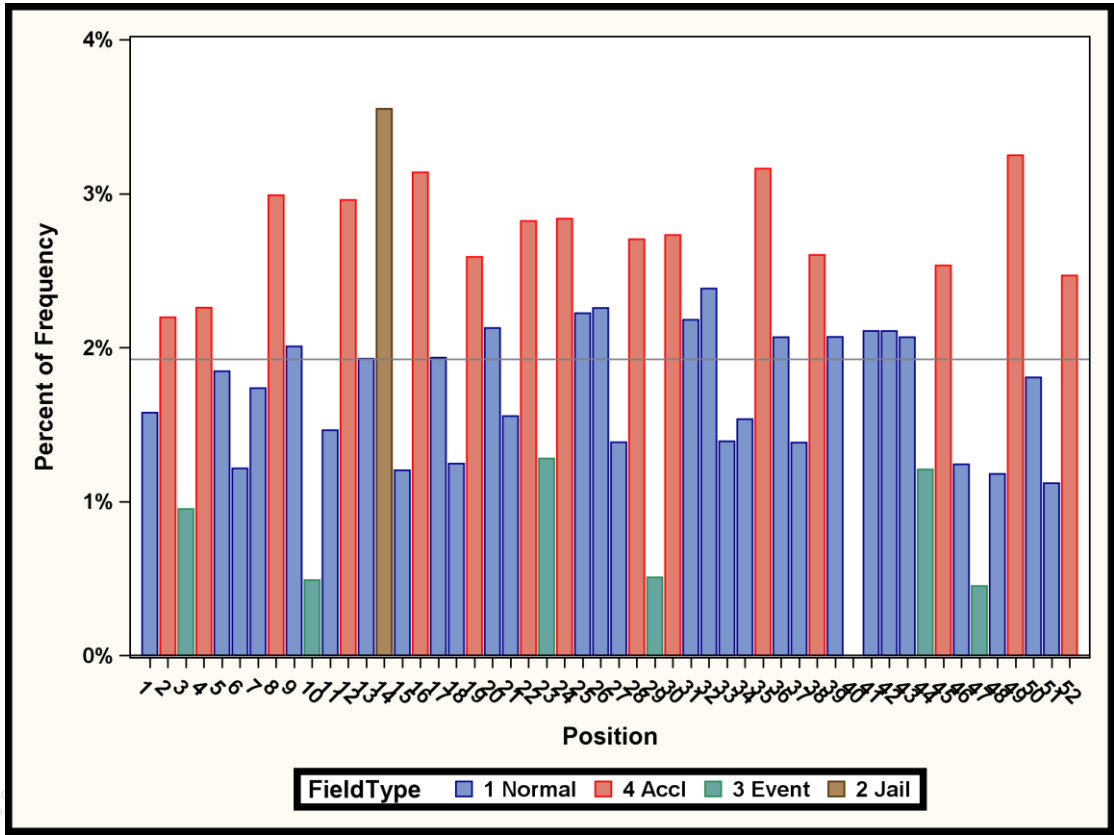


Effekt des Speedwürfels nach 70 Runden

„Dynamische Komponente“

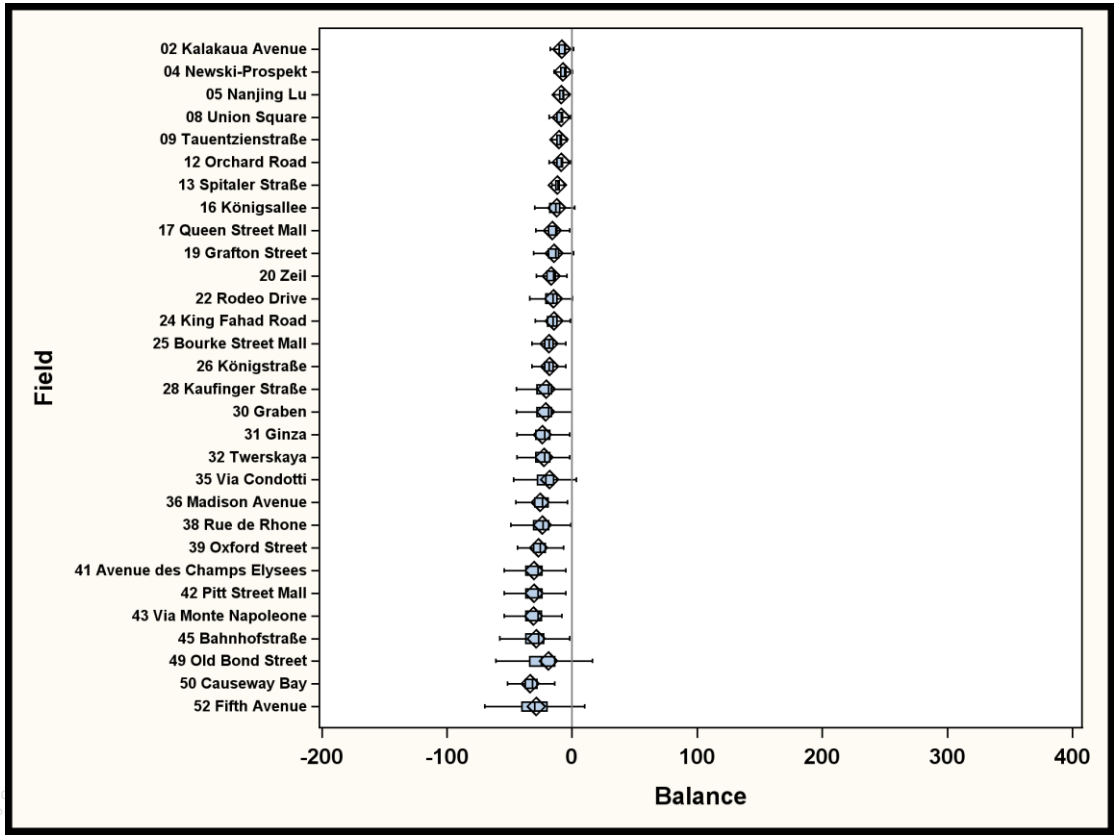
Effekt der Regel verändert sich im Laufe Spiels

Dynamische Komponente



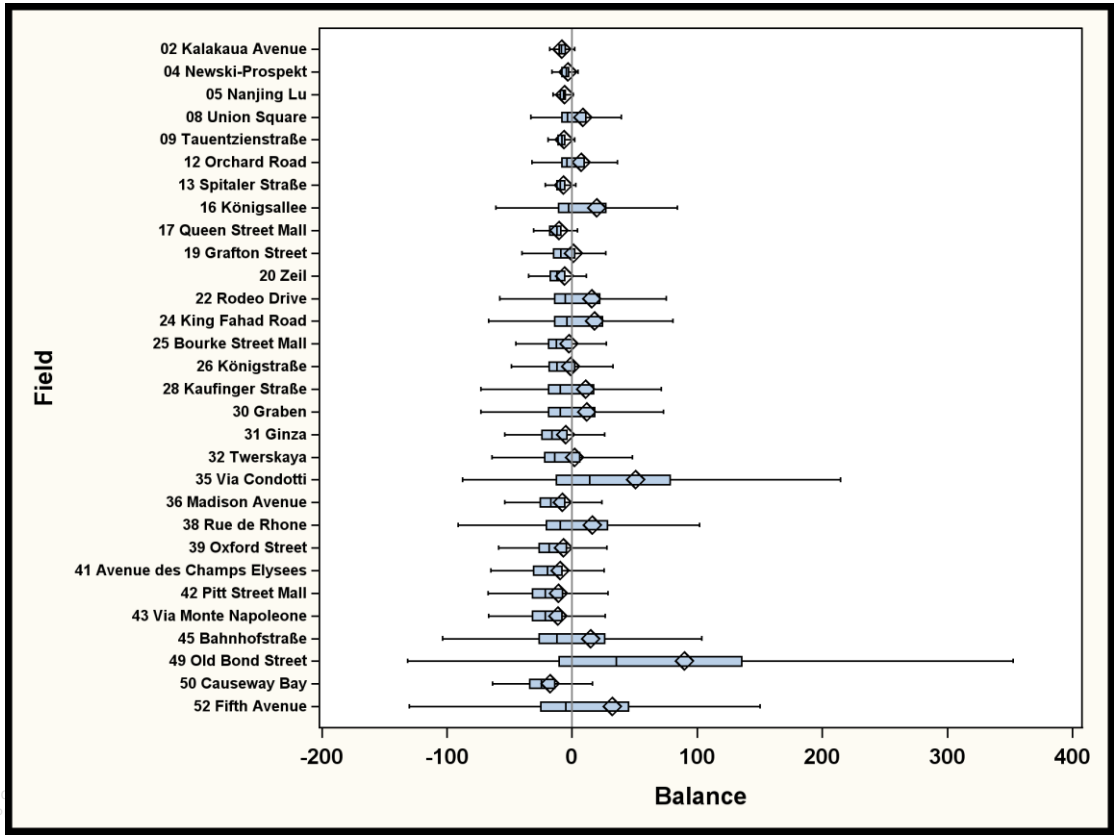
Profitabilitätsverteilung nach 40 Runden

Profitabilitätssimulation gibt Einblicke in die Verteilung des Profits pro Grundstück



Profitabilitätsverteilung nach 70 Runden

Die erwartete Dauer des Spiels hat Einfluss auf die Profitabilität der untschiedlichen Felder



Implementierung in SAS

Überblick

- Declare and Initialize
- Loop over Scenarios (Games)
  - Initialize Scenario
  - Loop over Rounds and Players
    - Generate Random Numbers
    - Follow Instructions
    - Generate Deterministic/Random Behaviour
    - Update Counts, Values, States
    - Output the Record
  - End Loop
- End Loop
- Prepare Analysis Data: Aggreg., Transpose, Enrich
- Calculate Output Statistics, Display Output

```
data Monopoly;
 array PlayerPos {4} PlayerPos1 - PlayerPos4;
   do Game = 1 to 10000;
     do Round = 1 to 70;
       do Player = 1 to 4;
          Dice1 = ceil(rand('Uniform')*6);
          if PlayerPos[Player]=40 then
                          PlayerPos[Player]=14;
          output;
       end;
     end;
   end;
run;
proc transpose data=Monopoly ...;run;
proc sgplot data=Monopoly_TP;
```

Verwendung von ARRAYs in einem SAS DATA Step

```
Array PlayerPos     {&players} PlayerPos1     - PlayerPos&players.      ;
Array PlayerBalance {&players} PlayerBalance1 - PlayerBalance&players.  ;
Array PlayerIncome  {&players} PlayerIncome1  - PlayerIncome&players.   ;
Array PlayerExpense {&players} PlayerExpense1 - PlayerExpense&players.  ;
Array Field         {52}       Field1   - Field52       ;
Array FieldSetup    {52} FieldSetup1    - FieldSetup52  ;
Array FieldRevenue  {52} FieldRevenue1  - FieldRevenue52;
Array FieldCost     {52} FieldCost1     - FieldCost52   ;
Array FieldBalance  {52} FieldBalance1  - FieldBalance52;
```

- **PLAYERPOS[2]** denotes the position of player 2 and refers to variable PLAYERPOS2
- **PLAYEREXPENSE[Player]** refers to the player expense variable for the respective player.
- **FIELDREVENUE[PLAYERPOS[PLAYER]]** refers to the revenue of that field, where the actual PLAYER is currently positioned.

Verwenden eines SAS Formats als Lookup Tabelle

	Field	M0	M1	M2	M3	M4
1	2	0.2	1	3	9	16
2	4	0.4	2	6	18	32
3	5	0.5	3	8	24	36
4	8	0.6	3	9	27	40
5	9	0.6	3	9	27	40
6	12	0.6	3	9	27	40
7	13	0.8	4	10	30	45
8	16	1	5	15	45	62
9	17	1	5	15	45	62
10	19	1	5	15	45	62
11	20	1.2	6	18	50	70

```
data k0;
 set Property_CostRevenue;
 fmtname = 'k0_';
 type = 'i';
 rename field=start k0=label;
run;
```

```
proc format cntlin=k0 library=work;
run;
```

Verwenden der Formate und Arrays

```
if Field[PlayerPos[Player]] = 0 then do;
      Field[PlayerPos[Player]]= Player;
      FieldCost[PlayerPos[Player]]= input(PlayerPos[Player],K0_.);
      FieldSetup[PlayerPos[Player]]= 0;
      PlayerExpense[Player]=
            PlayerExpense[Player] + input(PlayerPos[Player],K0_.);
end;
```

Bewegung der Spielfigur

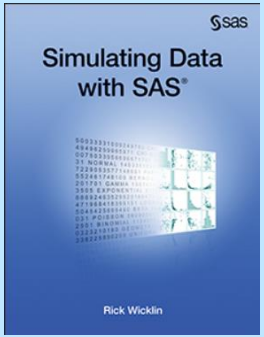
```
do Round = 1 to &Rounds;
                do Player = 1 to &players;
                        Dice1 = ceil(rand('Uniform')*6);
                       Dice2 = ceil(rand('Uniform')*6);
                        Dice3 = ceil(rand('Uniform')*6);
*** Dice3 shows a number Number that shall be added to the sum;
    if      Dice3 <= 3 then DiceSum = sum(Dice1,Dice2,Dice3);
    else                    DiceSum = sum(Dice1,Dice2);


        PlayerPos[Player] + DiceSum;
        PlayerPos[Player] = mod(PlayerPos[Player]-1,52)+1;
```

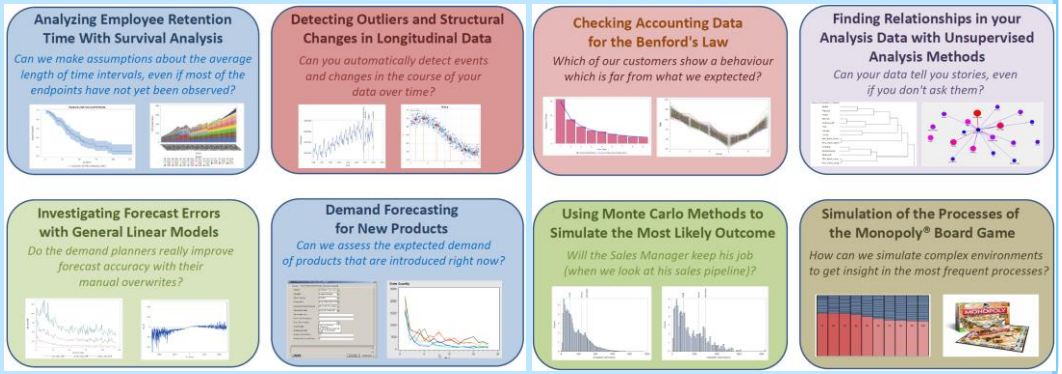
Links

- KSFE 2016: Simulationen und Mathematische Programmierung mit SAS, Gerhard Svolba

- Rick Wicklin:
  Simulating Data with SAS
  http://support.sas.com/publishing/authors/wicklin.html



- Gerhard Svolba: **Applying Data Science: Business Case Studies Using SAS**
  (SAS Press, expected 2017)



http://www.sascommunity.org/wiki/Applying_Data_Science_-_Business_Case_Studies_Using_SAS

See how data science can answer the questions your business faces!

Applying Data Science: Business Case Studies Using SAS®, by Gerhard Svolba, shows you the benefits of analytics, how to gain more insight into your data, and how to make better decisions. In eight entertaining and real-world case studies, Svolba combines data science and advanced analytics with business questions, illustrating them with data and SAS code.

The case studies range from a variety of fields, including performing head-count survival analysis for employee retention, forecasting the demand for new projects, using Monte Carlo simulation to understand outcome distribution, among other topics. The data science methods covered include Kaplan-Meier estimates, Cox Proportional Hazard Regression, ARIMA models, Poisson regression, imputation of missing values, variable clustering, and much more!

Written for business analysts, statisticians, data miners, data scientists, and SAS programmers, *Applying Data Science* bridges the gap between high-level, business-focused books that skimp on the details and technical books that only show SAS code with no business context.

GERHARD SVOLBA, PhD, is a principal solutions architect and analytic expert at SAS Institute Inc. based in Austria, where he specializes in application analytics and machine learning in different business and research domains. His project experience ranges from business and technical conceptual considerations to data preparation and analytic modeling across industries. He is the author of *Data Preparation for Analytics Using SAS®* and *Data Quality for Analytics using SAS®*, and he teaches the SAS training course "Building Analytic Data Marts." Gerhard likes to be in touch with customers and to exchange ideas about data science, machine learning, as well as data preparation and data quality in the context of analytics.

FREE DATA on the Web!
support.sas.com/authors

SAS Press

ISBN 978-1-60764-889-5

sas.com/books

sas

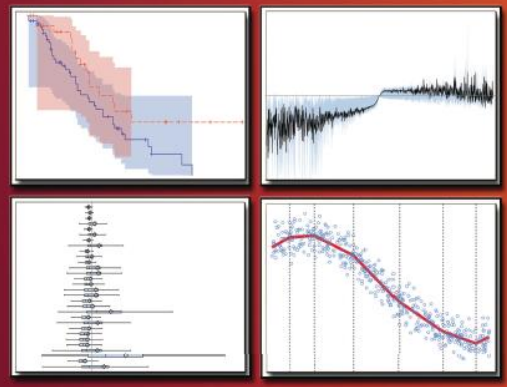
Applying Data Science
Business Case Studies Using SAS®

Svolba

sas

Applying Data Science

Business Case Studies Using SAS®



Gerhard Svolba

sas